

Práctica 1.5. RIP y BGP

Objetivos

En esta práctica se afianzan los conceptos elementales del encaminamiento. En particular, se estudia un protocolo de encaminamiento interior y otro exterior: RIP (*Routing Information Protocol*) y BGP (*Border Gateway Protocol*).

Existen muchas implementaciones de los protocolos de encaminamiento. En esta práctica vamos a utilizar Quagga, que actualmente implementa RIP (versiones 1 y 2), RIPng, OSPF, OSPFv3, IS-IS y BGP. Quagga está estructurado en diferentes servicios (uno para cada protocolo) controlados por un servicio central (Zebra) que hace de interfaz entre la tabla de reenvío del *kernel* y las tabla de encaminamiento de cada protocolo.

Todos los ficheros de configuración han de almacenarse en el directorio `/etc/quagga`. La sintaxis de estos ficheros es sencilla y está disponible en <http://quagga.net>. Revisar especialmente la correspondiente a RIP y BGP en <https://www.quagga.net/docs/quagga.html>. Además, en `/usr/share/doc/quagga-0.99.22.4` hay ficheros de ejemplo.



Activar el **portapapeles bidireccional** (menú Dispositivos) en las máquinas virtuales.

Usar la opción de Virtualbox (menú Ver) para realizar **capturas de pantalla**.

La **contraseña** del usuario `cursoresdes` es `cursoresdes`.

Contenidos

Parte I. Protocolo interior: RIP

- Preparación del entorno

- Configuración del protocolo RIP

Parte II. Protocolo exterior: BGP

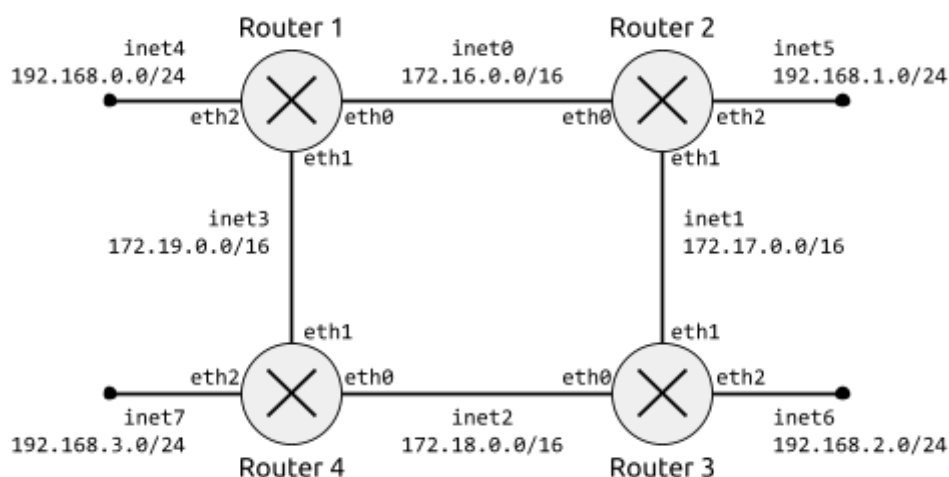
- Preparación del entorno

- Configuración del protocolo BGP

Parte I. Protocolo interior: RIP

Preparación del entorno

Configuraremos la topología de red que se muestra en la siguiente figura, donde cada encaminador (Router1...Router4) tiene tres interfaces, cada uno conectado a una red diferente.:



Al igual que en prácticas anteriores, usaremos la herramienta vtopo1 para construir automáticamente esta topología. A continuación se muestra el contenido del fichero de configuración de la topología:

```
netprefix inet
machine 1 0 0 1 3 2 4
machine 2 0 0 1 1 2 5
machine 3 0 2 1 1 2 6
machine 4 0 2 1 3 2 7
```

Para facilitar la configuración de las máquinas, la siguiente tabla muestra las direcciones de cada uno de sus interfaces de los encaminadores:

Máquina virtual	Interfaz	Dirección de red	Dirección IP
Router1	eth0	172.16.0.0/16	172.16.0.1
	eth1	172.19.0.0/16	172.19.0.1
	eth2	192.168.0.0/24	192.168.0.1
Router2	eth0	172.16.0.0/16	172.16.0.2
	eth1	172.17.0.0/16	172.17.0.2
	eth2	192.168.1.0/24	192.168.1.2
Router3	eth0	172.18.0.0/16	172.18.0.3
	eth1	172.17.0.0/16	172.17.0.3
	eth2	192.168.2.0/24	192.168.2.3
Router4	eth0	172.18.0.0/16	172.18.0.4
	eth1	172.19.0.0/16	172.19.0.4
	eth2	192.168.3.0/24	192.168.3.4

Configurar todos los encaminadores según la figura y tabla anterior. Además, activar el reenvío de paquetes IPv4 igual que en la práctica 1.1. Después, comprobar:

- Que los encaminadores adyacentes son alcanzables, por ejemplo, Router1 puede hacer *ping* a Router2 y Router4.
- Que la tabla de reenvío de cada encaminador es la correcta e incluye una entrada para cada una de las tres redes a las que está conectado.

En VM1:

```
sudo ip address add 172.16.0.1/16 dev eth0
sudo ip address add 172.19.0.1/16 dev eth1
sudo ip address add 192.168.0.1/24 dev eth2
```

```

En VM2:
sudo ip address add 172.16.0.2/16 dev eth0
sudo ip address add 172.17.0.2/16 dev eth1
sudo ip address add 192.168.1.2/24 dev eth2
En VM3:
sudo ip address add 172.18.0.3/16 dev eth0
sudo ip address add 172.17.0.3/16 dev eth1
sudo ip address add 192.168.2.2/24 dev eth2
En VM4:
sudo ip address add 172.18.0.4/16 dev eth0
sudo ip address add 172.19.0.4/16 dev eth1
sudo ip address add 192.168.3.4/24 dev eth2

sudo ip link set eth0 up
sudo ip link set eth1 up
sudo ip link set eth2 up

sudo sysctl net.ipv4.ip_forward=1

ping -c 1 172.16.0.2
PING 172.16.0.2 (172.16.0.2) 56(84) bytes of data.
64 bytes from 172.16.0.2: icmp_seq=1 ttl=64 time=0.808 ms

--- 172.16.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.808/0.808/0.808/0.000 ms

ping -c 1 172.19.0.4
PING 172.19.0.4 (172.19.0.4) 56(84) bytes of data.
64 bytes from 172.19.0.4: icmp_seq=1 ttl=64 time=1.65 ms

--- 172.19.0.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.650/1.650/1.650/0.000 ms

En VM1:
sudo ip route
172.16.0.0/16 dev eth0 proto kernel scope link src 172.16.0.1
172.19.0.0/16 dev eth1 proto kernel scope link src 172.19.0.1
192.168.0.0/24 dev eth2 proto kernel scope link src 192.168.0.1

```

Configuración del protocolo RIP

Ejercicio 1. Configurar RIP en todos los encaminadores para que intercambien información:

- Crear un fichero `ripd.conf` en `/etc/quagga` con el contenido que se muestra a continuación.
- Iniciar el servicio RIP (y Zebra) con `service ripd start`.

Contenido del fichero `/etc/quagga/ripd.conf`:

```

sudo nano /etc/quagga/ripd.conf

# Activar el encaminamiento por RIP
router rip
# Definir la versión del protocolo que se usará
version 2

```

```
# Habilitar información de encaminamiento en redes asociadas a los interfaces
network eth0
network eth1
network eth2

service ripd start
```

Ejercicio 2. Consultar la tabla de encaminamiento de RIP y de Zebra en cada encaminador con el comando vtysh (sudo vtysh -c "show ip rip" y sudo vtysh -c "show ip route"). Comprobar también la tabla de reenvío de IPv4 con el comando ip (ip route).

Copia los comandos usados y su salida.

En VM1:

```
sudo vtysh -c "show ip rip"
```

Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP

Sub-codes:

(n) - normal, (s) - static, (d) - default, (r) - redistribute,
(i) - interface

	Network	Next Hop	Metric From	Tag Time
C(i)	172.16.0.0/16	0.0.0.0	1 self	0
R(n)	172.17.0.0/16	172.16.0.2	2 172.16.0.2	0 02:35
R(n)	172.18.0.0/16	172.19.0.4	2 172.19.0.4	0 02:34
C(i)	172.19.0.0/16	0.0.0.0	1 self	0
C(i)	192.168.0.0/24	0.0.0.0	1 self	0
R(n)	192.168.1.0/24	172.16.0.2	2 172.16.0.2	0 02:35
R(n)	192.168.2.0/24	172.16.0.2	3 172.16.0.2	0 02:35
R(n)	192.168.3.0/24	172.19.0.4	2 172.19.0.4	0 02:34

```
sudo vtysh -c "show ip route"
```

Codes: K - kernel route, C - connected, S - static, R - RIP,
O - OSPF, I - IS-IS, B - BGP, A - Babel,
> - selected route, * - FIB route

```
C>* 127.0.0.0/8 is directly connected, lo
C>* 172.16.0.0/16 is directly connected, eth0
R>* 172.17.0.0/16 [120/2] via 172.16.0.2, eth0, 00:02:24
R>* 172.18.0.0/16 [120/2] via 172.19.0.4, eth1, 00:02:08
C>* 172.19.0.0/16 is directly connected, eth1
C>* 192.168.0.0/24 is directly connected, eth2
R>* 192.168.1.0/24 [120/2] via 172.16.0.2, eth0, 00:02:24
R>* 192.168.2.0/24 [120/3] via 172.16.0.2, eth0, 00:02:16
R>* 192.168.3.0/24 [120/2] via 172.19.0.4, eth1, 00:02:08
```

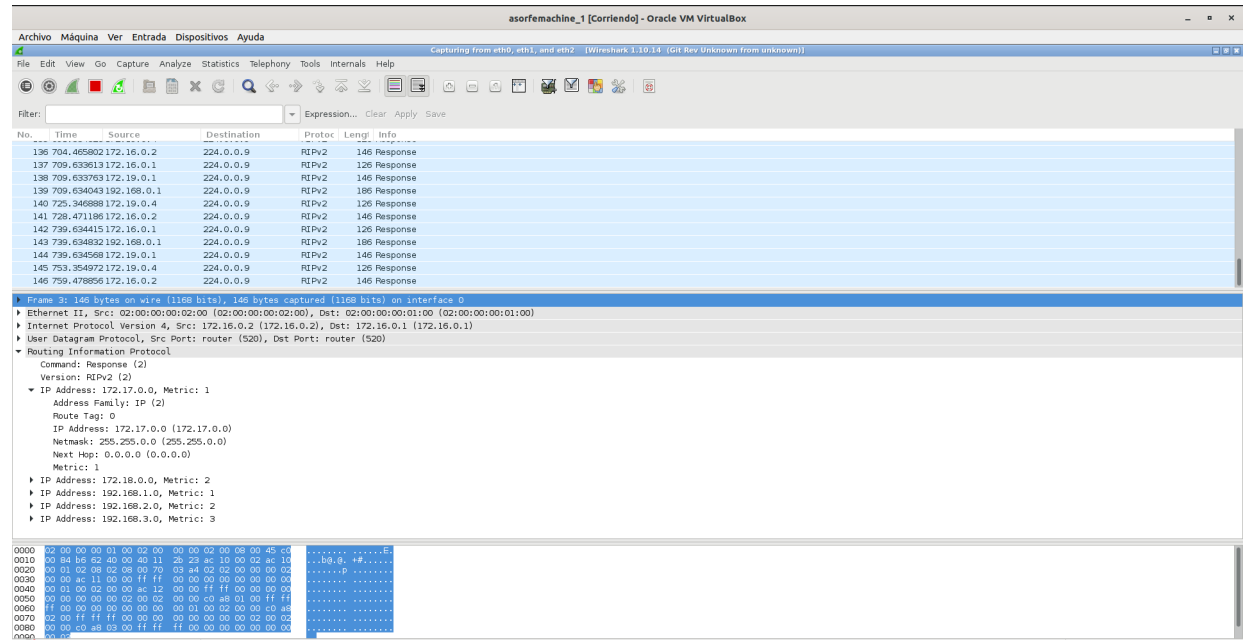
```
sudo ip route
```

```
172.16.0.0/16 dev eth0 proto kernel scope link src 172.16.0.1
172.17.0.0/16 via 172.16.0.2 dev eth0 proto zebra metric 2
172.18.0.0/16 via 172.19.0.4 dev eth1 proto zebra metric 2
172.19.0.0/16 dev eth1 proto kernel scope link src 172.19.0.1
192.168.0.0/24 dev eth2 proto kernel scope link src 192.168.0.1
192.168.1.0/24 via 172.16.0.2 dev eth0 proto zebra metric 2
192.168.2.0/24 via 172.16.0.2 dev eth0 proto zebra metric 3
192.168.3.0/24 via 172.19.0.4 dev eth1 proto zebra metric 2
```

Ejercicio 3. Con la herramienta wireshark, estudiar los mensajes RIP intercambiados, en particular:

- Encapsulado.
- Direcciones origen y destino.
- Campo de versión.
- Información para cada ruta: dirección de red, máscara de red, siguiente salto y distancia.

Copy una captura de pantalla de Wireshark con mensajes RIP mostrando el formato de uno de ellos. En VM1:



Encapsulado : udp

Version: RIPv2

Direcciones:

172.16.0.1 - 172.16.0.2

172.19.0.1 - 172.19.0.4

224.0.0.9 protocolo estándar de direcciones multicast, reservado para speaking routers

Dirección de red: 172.17.0.0

Máscara de red 255.255.0.0

Siguiente salto 0.0.0.0 (la tabla esta ya completa)

Distancia 1

En las de request aparece dir de red, máscara next hop a 0.0.0.0 y métrica a 16

Resumiendo en la información aparecen los datos de las tablas del ejercicio anterior

Ejercicio 4. Eliminar el enlace entre Router1 y Router4 (por ejemplo, desactivando el interfaz eth1 en Router4). Comprobar que Router1 deja de recibir los anuncios de Router4 y que, pasados aproximadamente 3 minutos (valor de *timeout* por defecto para las rutas), ha reajustado su tabla.

Copia los comandos usados y su salida.

En VM4:

sudo ip link set eth1 down


```
$ telnet localhost ripd
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.20.1)
Copyright © 1996-2005 Kunihiro Ishiguro, et al.
User Access Verification

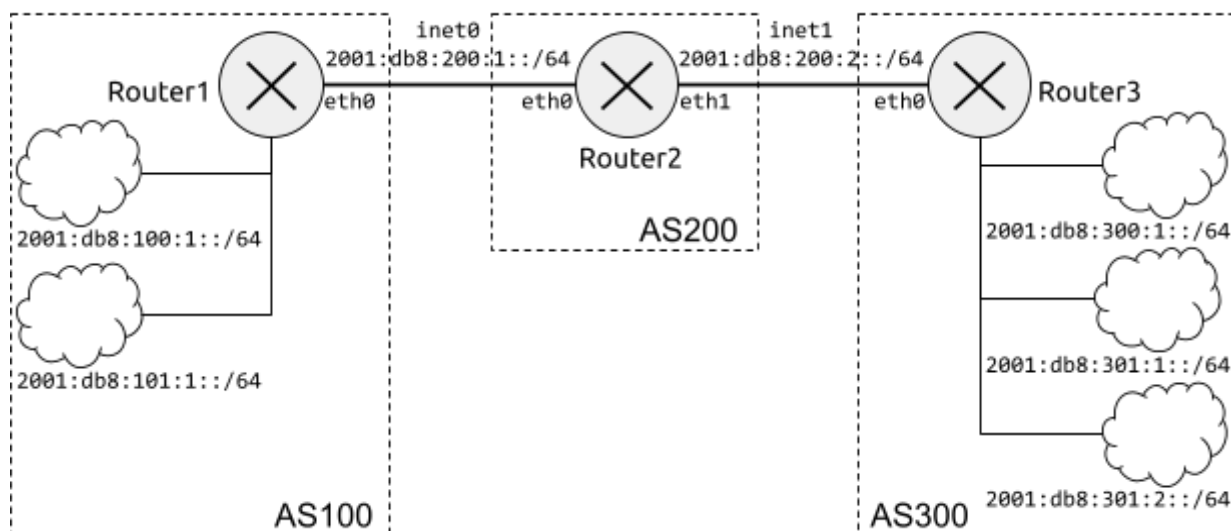
Password: asor
localhost.localdomain> enable
localhost.localdomain# configure terminal
localhost.localdomain(config)# router rip
localhost.localdomain(config-router)# version 2
localhost.localdomain(config-router)# network eth0
localhost.localdomain(config-router)# write
Configuration saved to /etc/quagga/ripd.conf
localhost.localdomain(config-router)# exit
localhost.localdomain(config)# exit
localhost.localdomain# show running-config
Current configuration:
!
password asor
!
router rip
  version 2
  network eth0
!
line vty
!
end
localhost.localdomain# write
Configuration saved to /etc/quagga/ripd.conf
localhost.localdomain# exit
```

Nota: Para poder escribir la configuración en ripd.conf, el usuario quagga debe tener los permisos adecuados sobre el fichero. Para cambiar el propietario del fichero, ejecutar el comando `chown quagga:quagga /etc/quagga/ripd.conf`.

Parte II. Protocolo exterior: BGP

Preparación del entorno

Configuraremos la topología de red con 3 AS, siendo uno de ellos el proveedor de los otros dos:



Nota: El prefijo 2001:db8::/32 está reservado para documentación y ejemplos (RFC 3849).

Crearemos esta topología (sin las redes internas de los AS) con la herramienta vtopo1 y el siguiente fichero:

```
netprefix inet
machine 1 0 0
machine 2 0 0 1 1
machine 3 0 1
```

Para facilitar la configuración de las máquinas, la siguiente tabla muestra las direcciones de cada uno de los interfaces de los encaminadores:

Máquina virtual	Interfaz	Dirección de red	Dirección IP
Router1	eth0	2001:db8:200:1::/64	2001:db8:200:1::1
Router2	eth0	2001:db8:200:1::/64	2001:db8:200:1::2
	eth1	2001:db8:200:2::/64	2001:db8:200:2::2
Router3	eth0	2001:db8:200:2::/64	2001:db8:200:2::3

Configurar los encaminadores según se muestra en la figura anterior. Debe comprobarse la conectividad entre máquinas adyacentes.

```
En VM1:
ip link set eth0 up
ip addr add 2001:db8:200:1::1/64 dev eth0
En VM2:
ip link set eth0 up
ip link set eth1 up
```



```
ip addr add 2001:db8:200:1::2/64 dev eth0
ip addr add 2001:db8:200:2::2/64 dev eth1
En VM3:
ip link set eth0 up
ip addr add 2001:db8:200:2::3/64 dev eth0
```

Configuración del protocolo BGP

Ejercicio 6. Consultar la documentación de las clases de teoría para determinar el tipo de AS (*stub*, *multihomed* o *transit*) y los prefijos de red que debe anunciar. Recordar que el prefijo global de encaminamiento es de 48 bits y que los prefijos anunciados deben agregarse al máximo.

Número de AS	Tipo	Prefijos agregados
AS100	Stub	2001:db8:100::/47
AS200	Transit	
AS100	Stub	2001:db8:300::/47

Ejercicio 7. Configurar BGP en los encaminadores para que intercambien información:

- Crear un fichero `bgpd.conf` en `/etc/quagga` usando como referencia el que se muestra a continuación.
- Iniciar el servicio BGP (y Zebra) con `service bgpd start`.

Por ejemplo, el contenido del fichero `/etc/quagga/bgpd.conf` de Router1 en el AS 100 sería:

```
En VM1: nano /etc/quagga/bgpd.conf
# Activar el encaminamiento BGP en el AS 100
router bgp 100
# Establecer el identificador de encaminador BGP
bgp router-id 0.0.0.1
# Añadir el encaminador BGP vecino en el AS 200
neighbor 2001:db8:200:1::2 remote-as 200
# Empezar a trabajar con direcciones IPv6
address-family ipv6
# Anunciar un prefijo de red agregado
network 2001:db8:100::/47
# Activar IPv6 en el encaminador BGP vecino
neighbor 2001:db8:200:1::2 activate
# Dejar de trabajar con direcciones IPv6
Exit-address-family
```

```
En VM2: nano /etc/quagga/bgpd.conf
# Activar el encaminamiento BGP en el AS 200
router bgp 200
# Establecer el identificador de encaminador BGP
bgp router-id 0.0.0.2
# Añadir el encaminador BGP vecino en el AS 100
neighbor 2001:db8:200:1::1 remote-as 100
# Añadir el encaminador BGP vecino en el AS 300
neighbor 2001:db8:200:2::3 remote-as 300
# Empezar a trabajar con direcciones IPv6
address-family ipv6
# Activar IPv6 en el encaminador BGP vecino
```

```

neighbor 2001:db8:200:1::1 activate
neighbor 2001:db8:200:2::3 activate
# Dejar de trabajar con direcciones IPv6
Exit-address-family

En VM3: nano /etc/quagga/bgpd.conf
# Activar el encaminamiento BGP en el AS 100
router bgp 300
# Establecer el identificador de encaminador BGP
bgp router-id 0.0.0.3
# Añadir el encaminador BGP vecino en el AS 200
neighbor 2001:db8:200:2::2 remote-as 200
# Empezar a trabajar con direcciones IPv6
address-family ipv6
# Anunciar un prefijo de red agregado
network 2001:db8:300::/47
# Activar IPv6 en el encaminador BGP vecino
neighbor 2001:db8:200:2::2 activate
# Dejar de trabajar con direcciones IPv6
Exit-address-family

service bgpd start

```

Ejercicio 8. Consultar la tabla de encaminamiento de BGP y de Zebra en cada encaminador con el comando vtysh (sudo vtysh -c "show ipv6 bgp" y sudo vtysh -c "show ipv6 route"). Comprobar también la tabla de reenvío de IPv6 con el comando ip (ip -6 route).

Copia los comandos usados y su salida.

En VM1:

vtysh -c "show ipv6 bgp"

BGP table version is 0, local router ID is 0.0.0.1

*Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, R Removed*

Origin codes: i - IGP, e - EGP, ? - incomplete

```

Network      Next Hop      Metric LocPrf Weight Path
*> 2001:db8:100::/47
           ::          0      32768 i
*> 2001:db8:300::/47
           2001:db8:200:1::2
                        0 200 300 i

```

Total number of prefixes 2

vtysh -c "show ipv6 route"

Codes: K - kernel route, C - connected, S - static, R - RIPv6,

O - OSPFv6, I - IS-IS, B - BGP, A - Babel,

*> - selected route, * - FIB route*

C>::1/128 is directly connected, lo*

C> 2001:db8:200:1::/64 is directly connected, eth0*

B> 2001:db8:300::/47 [20/0] via fe80::a00:27ff:feea:9bf2, eth0, 00:01:32*

C> fe80::/64 is directly connected, eth0*

ip -6 route

```
unreachable ::/96 dev lo metric 1024 error -113 pref medium
unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -113 pref medium
2001:db8:200:1::/64 dev eth0 proto kernel metric 256 pref medium
2001:db8:300::/47 via fe80::a00:27ff:feea:9bf2 dev eth0 proto zebra metric 1024 pref medium
unreachable 2002:a00::/24 dev lo metric 1024 error -113 pref medium
unreachable 2002:7f00::/24 dev lo metric 1024 error -113 pref medium
unreachable 2002:a9fe::/32 dev lo metric 1024 error -113 pref medium
unreachable 2002:ac10::/28 dev lo metric 1024 error -113 pref medium
unreachable 2002:c0a8::/32 dev lo metric 1024 error -113 pref medium
unreachable 2002:e000::/19 dev lo metric 1024 error -113 pref medium
unreachable 3ffe:ffff::/32 dev lo metric 1024 error -113 pref medium
fe80::/64 dev eth0 proto kernel metric 256 pref medium
```

En VM2:

vttysh -c "show ipv6 bgp"

BGP table version is 0, local router ID is 0.0.0.2

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, R Removed

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2001:db8:100::/47					
	2001:db8:200:1::1				
	0	0	100	i	
*> 2001:db8:300::/47					
	2001:db8:200:2::3				
	0	0	300	i	

Total number of prefixes 2

vttysh -c "show ipv6 route"

Codes: K - kernel route, C - connected, S - static, R - RIPng,

O - OSPFv6, I - IS-IS, B - BGP, A - Babel,

> - selected route, * - FIB route

C>* ::1/128 is directly connected, lo

B>* 2001:db8:100::/47 [20/0] via fe80::a00:27ff:feb8:ad92, eth0, 00:06:37

C>* 2001:db8:200:1::/64 is directly connected, eth0

C>* 2001:db8:200:2::/64 is directly connected, eth1

B>* 2001:db8:300::/47 [20/0] via fe80::a00:27ff:fe49:1476, eth1, 00:06:11

C>* fd00:0:0:a::/64 is directly connected, eth1

C * fe80::/64 is directly connected, eth1

C>* fe80::/64 is directly connected, eth0

ip -6 route

```
unreachable ::/96 dev lo metric 1024 error -113 pref medium
unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -113 pref medium
2001:db8:100::/47 via fe80::a00:27ff:feb8:ad92 dev eth0 proto zebra metric 1024 pref medium
2001:db8:200:1::/64 dev eth0 proto kernel metric 256 pref medium
2001:db8:200:2::/64 dev eth1 proto kernel metric 256 pref medium
2001:db8:300::/47 via fe80::a00:27ff:fe49:1476 dev eth1 proto zebra metric 1024 pref medium
unreachable 2002:a00::/24 dev lo metric 1024 error -113 pref medium
unreachable 2002:7f00::/24 dev lo metric 1024 error -113 pref medium
unreachable 2002:a9fe::/32 dev lo metric 1024 error -113 pref medium
unreachable 2002:ac10::/28 dev lo metric 1024 error -113 pref medium
unreachable 2002:c0a8::/32 dev lo metric 1024 error -113 pref medium
```

```

unreachable 2002:e000::/19 dev lo metric 1024 error -113 pref medium
unreachable 3ffe:ffff::/32 dev lo metric 1024 error -113 pref medium
fd00:0:0:a::/64 dev eth1 proto kernel metric 256 expires 2591704sec pref medium
fe80::/64 dev eth0 proto kernel metric 256 pref medium
fe80::/64 dev eth1 proto kernel metric 256 pref medium

```

En VM2:

vtysh -c "show ipv6 bgp"

BGP table version is 0, local router ID is 0.0.0.3

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, R Removed

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2001:db8:100::/47	2001:db8:200:2::2	0	200	100	i
*> 2001:db8:300::/47	::	0	32768		i

Total number of prefixes 2

vtysh -c "show ipv6 route"

Codes: K - kernel route, C - connected, S - static, R - RIPng,

O - OSPFv6, I - IS-IS, B - BGP, A - Babel,

> - selected route, * - FIB route

C>* ::1/128 is directly connected, lo

B>* 2001:db8:100::/47 [20/0] via fe80::a00:27ff:fe73:c87, eth0, 00:07:30

C>* 2001:db8:200:2::/64 is directly connected, eth0

C>* fe80::/64 is directly connected, eth0

ip -6 route

```

unreachable ::/96 dev lo metric 1024 error -113 pref medium
unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -113 pref medium
2001:db8:100::/47 via fe80::a00:27ff:fe73:c87 dev eth0 proto zebra metric 1024 pref medium
2001:db8:200:2::/64 dev eth0 proto kernel metric 256 pref medium
unreachable 2002:a00::/24 dev lo metric 1024 error -113 pref medium
unreachable 2002:7f00::/24 dev lo metric 1024 error -113 pref medium
unreachable 2002:a9fe::/32 dev lo metric 1024 error -113 pref medium
unreachable 2002:ac10::/28 dev lo metric 1024 error -113 pref medium
unreachable 2002:c0a8::/32 dev lo metric 1024 error -113 pref medium
unreachable 2002:e000::/19 dev lo metric 1024 error -113 pref medium
unreachable 3ffe:ffff::/32 dev lo metric 1024 error -113 pref medium
fe80::/64 dev eth0 proto kernel metric 256 pref medium

```

Ejercicio 9. Con ayuda de la herramienta wireshark, estudiar los mensajes BGP intercambiados (OPEN, KEEPALIVE y UPDATE).

Copia una captura de pantalla de Wireshark con mensajes BGP mostrando el formato del mensaje UPDATE.

Captura de Router 2

Router 2

Markers: ffffffff...

Length: 88

Type: UPDATE Message (2)

Unfeasible routes Length: 0 bytes

Total path attribute Length: 65 bytes

Path attributes

- ORIGIN: IGP (4 bytes)
- AS_PATH: 200 300 (14 bytes)
 - Flags: 0x50 (Well-known, Transitive, Complete, Extended Length)
 - Type code: AS_PATH (2)
 - Length: 10 bytes
 - AS path: 200 300
- MP_REACH_NLRI (47 bytes)
 - Flags: 0xB0 (Optional, Non-transitive, Complete)
 - Type code: MP_REACH_NLRI (14)
 - Length: 44 bytes
 - Address family: IPv6 (2)
 - Subsequent address family identifier: Unicast (1)
 - Next hop network address (32 bytes)
 - Subnetwork points of attachment: 0
 - Network Layer reachability information (7 bytes)
 - 2001:DB8:3005::7:47

0000 38 00 27 b8 ad 32 08 00 27 ea 8b f2 85 dd 5c 00

0010 00 00 00 78 06 01 20 01 04 b8 02 00 00 01 00 00

0020 00 00 00 00 00 02 20 01 04 b8 02 00 00 01 00 00

0030 00 00 00 00 01 b7 40 00 b3 4b b1 0c 55 1e 4d 00

0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Frame (Frame), 174 bytes Packets: 604 - Displayed: 604 (100.0%) Profile: Default

Campos:

Longitud, tipo, long unfeasible routes, path attribute (origin, as_path, mp_reach_nlri (redes alcanzables, next hops, longitud, tipo))